Disentangling Optimization Geometry via Hierarchical Polar Adapters for Class-Incremental Learning

Dat Quang Mac[1] **Thanh Hai Dang**[2] **Duc-Trong Le**[2] **Quynh-Trang Pham Thi**[2*]
[1]Independent Researcher
[2]VNU University of Engineering and Technology, Hanoi, Vietnam
dat.q.mac@gmail.com, {haidt, trongld, trangptq}@vnu.edu.vn

Abstract

Class-incremental learning (CIL) requires models to acquire new knowledge while preserving previously learned representations, yet sequential adaptation often leads to catastrophic forgetting. This challenge is particularly pronounced in strictly exemplar-free settings, where past training samples cannot be revisited. We propose **HiPo** (**Hi**erarchical **Po**lar Adapters), a parameter-efficient continual learning framework that addresses task interference through Fourier-projected adaptation and hierarchical knowledge consolidation. First, HiPo employs a fixed orthonormal Fourier transform to distribute localized adapter updates across feature coordinates, combined with complex block-diagonal adapters parameterized in magnitude and phase for flexible adaptation. Second, a fixed-depth hierarchical memory architecture separates a plastic working tier from deeper long-term memory tiers, enabling knowledge accumulation without increasing the adapter parameter budget across tasks. Third, an adaptive Snapshot Fisher mechanism guides continuous, importance-weighted Cartesian transfers between memory tiers, supporting gradual knowledge consolidation while approximately preserving the effective adapter representation. Extensive experiments across standard CIL benchmarks demonstrate competitive performance, with particularly strong robustness to longer task sequences and challenging distribution shifts. Ablation studies highlight Fourier-based spatial dispersion as the primary contributor to reducing interference, while polar parameterization and hierarchical consolidation provide complementary, setting-dependent improvements. These findings demonstrate the effectiveness of combining distributed feature adaptation with structured knowledge consolidation for parameter-efficient continual learning.

1 Introduction

Continual learning aims to enable neural networks to acquire new knowledge without forgetting previously learned information. However, sequential adaptation often leads to catastrophic forgetting, as updates for new tasks interfere with existing representations (30; 39). This challenge is particularly pronounced in class-incremental learning (CIL), where models must recognize an expanding set of classes without task identity at inference. In strictly exemplar-free settings, the inability to revisit past training samples makes balancing knowledge retention and adaptation especially difficult (32).

Pretrained models and parameter-efficient adaptation offer promising solutions by preserving general-purpose features while updating a small set of trainable parameters (15). Nevertheless, parameter efficiency alone does not prevent interference between sequential updates. Existing approaches often address forgetting through parameter regularization, task-specific components, or constrained optimization subspaces (42; 23; 45). While effective, these strategies may restrict adaptation capacity or require additional parameters as tasks accumulate. This motivates an alternative approach

*Corresponding author.

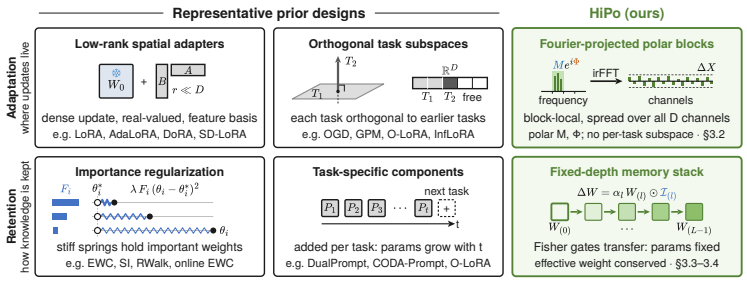


Figure 1: **HiPo versus representative prior designs.** *Top:* prior adapters use dense low-rank updates in the feature basis or confine each task to an orthogonal subspace, which uses up dimensions; HiPo spreads block-local polar updates over all channels through a fixed Fourier basis (§3.2). *Bottom:* prior methods penalize changes to important weights or add parameters for each task; HiPo keeps a fixed-depth stack and uses Fisher importance to gate transfers between tiers (§3.3–3.4).

that distributes updates across feature coordinates rather than enforcing strict separation between tasks. Combined with an efficient mechanism for retaining and consolidating knowledge, such a representation could support continued adaptation without increasing model capacity for every new task.

We propose **HiPo** (**Hi**erarchical **Po**lar Adapters), a parameter-efficient framework for strictly exemplar-free CIL that combines Fourier-projected adaptation with hierarchical knowledge consolidation. First, HiPo applies a fixed orthonormal Fourier transform to mix feature channels, allowing localized adapter updates to spread across spatial coordinates after reconstruction. Complex block-diagonal adapters are then parameterized in magnitude and phase to provide separate coordinates for optimization. Second, a fixed-depth memory hierarchy separates a plastic working-memory tier from deeper long-term memory tiers, maintaining a constant adapter parameter budget across tasks. Finally, periodically estimated Snapshot Fisher scores guide continuous, importance-weighted Cartesian transfers between tiers. These transfers preserve the effective adapter weights when the superposition weights are held fixed and numerical safeguards remain inactive, enabling knowledge redistribution without immediate changes to the effective transformation.

Our main contributions are summarized as follows:

- **Fourier-projected polar adaptation.** We introduce complex block-diagonal adapters that combine parameter-free Fourier mixing with magnitude–phase parameterization. We analyze the spatial dispersion of localized updates and evaluate its role in mitigating task interference.
- **Hierarchical memory architecture.** We develop a fixed-depth memory stack that separates working-memory adaptation from longer-term knowledge storage, enabling sequential learning without task-dependent growth in adapter parameters.
- **Importance-driven consolidation.** We propose an adaptive Snapshot Fisher mechanism that guides continuous Cartesian transfers between memory tiers while preserving the effective adapter weights under specified conditions.

We evaluate HiPo on four strictly exemplar-free CIL benchmarks. Experimental results demonstrate competitive performance, particularly under longer task sequences and challenging distribution shifts. Ablation studies identify Fourier-based spatial dispersion as the primary contributor to long-sequence robustness, while polar parameterization and hierarchical consolidation provide complementary, setting-dependent benefits. These findings support the effectiveness of combining distributed feature adaptation with structured knowledge consolidation for parameter-efficient continual learning.

2 Related work

2.1 Continual Learning and Knowledge Consolidation

Class-incremental learning (CIL) aims to mitigate catastrophic forgetting when learning sequential tasks (30; 39). Traditional approaches rely on episodic rehearsal (32; 27) or parameter regularization

(17; 46). More recent methods leverage frozen pretrained backbones to preserve general-purpose representations. Prompt-based approaches, including L2P, DualPrompt, and CODA-Prompt, adapt representations through learned prompts (44; 43; 36), while subspace-based methods constrain parameter updates to reduce interference (9; 34; 42). Random projection methods, such as RanPAC (31), offer another approach to maintaining feature separability. However, balancing adaptation capacity, knowledge retention, and parameter efficiency remains challenging, particularly over long task sequences. Building on the almost-orthogonality perspective of BiLoRA (48), HiPo explores distributed feature adaptation without requiring strictly orthogonal task-specific subspaces or an expanding adapter architecture.

Knowledge consolidation provides a complementary perspective on continual learning. Complementary Learning Systems theory describes how learning and retention can operate at different timescales (29; 20), motivating hierarchical architectures such as Nested Learning (3). Separately, Fisher-information-based methods estimate parameter importance to protect previously acquired knowledge (1; 4; 17; 33; 35). Inspired by these directions, HiPo combines a fixed-depth memory hierarchy with Snapshot Fisher-guided consolidation. Rather than applying Fisher estimates as regularization penalties, HiPo uses them to control continuous transfers between memory tiers, without assuming that the hierarchy or importance estimates alone guarantee resistance to forgetting (Figure 1, bottom).

2.2 Structured Adaptation and Frequency-Domain Representations

Parameter-efficient fine-tuning (PEFT) enables model adaptation with a limited number of trainable parameters. Representative methods include low-rank adaptation techniques such as LoRA, QLoRA, and AdaLoRA (15; 6; 47), as well as parameter-efficient scaling methods such as IA^3 and VeRA (24; 18). Other approaches, including DoRA, separate magnitude and directional components to improve adaptation (26). These methods demonstrate the effectiveness of structured parameterization, but their use of spatial-domain representations differs from HiPo's complex-valued polar adapters.

Frequency-domain representations offer an alternative way to transform and mix features. Fourier-based architectures, including FNO, FNet, and AFNO, demonstrate the utility of spectral transformations for efficient representation learning (22; 21; 10). Related work on complex-valued neural networks investigates structured complex transformations and parameterizations (2; 38), while spatial wavelets have also been explored for continual learning (25). In contrast to these approaches, HiPo uses a fixed orthonormal Fourier transform to distribute localized adapter updates across feature coordinates (Figure 1, top). It combines this projection with complex block-diagonal polar adaptation and hierarchical knowledge consolidation, bringing these components together in a parameter-efficient framework for strictly exemplar-free CIL.

3 Methodology

3.1 Formulation

We formulate our approach within the strictly exemplar-free Class-Incremental Learning (CIL) paradigm. A model $f(\cdot;\Theta)$ sequentially learns N disjoint tasks, $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, and must classify samples across the expanding global label space $\mathcal{Y}_{1:t}$ without a task oracle or buffered episodic data. To decouple generalized feature extraction from sequential adaptation, we partition Θ into a permanently frozen pre-trained backbone ϕ and a set of parameter-efficient incremental adapters θ. Adapter updates may interfere with features used by previous tasks. HiPo studies whether a fixed dense projection and a reusable memory stack reduce per-coordinate interference without imposing an exact task-by-task orthogonality constraint.

The method has three components: Fourier-projected polar block-diagonal adaptation (§3.2), a fixed-depth hierarchical memory stack (§3.3), and Snapshot Fisher-guided Cartesian consolidation (§3.4). Appendix A provides the training and transfer algorithms.

3.2 Fourier-Projected Polar Block-Diagonal Adaptation

Strictly orthogonal task subspaces can impose a capacity constraint as the number of tasks increases. Building on the almost-orthogonality perspective of BiLoRA (48), we employ fixed dense mixing

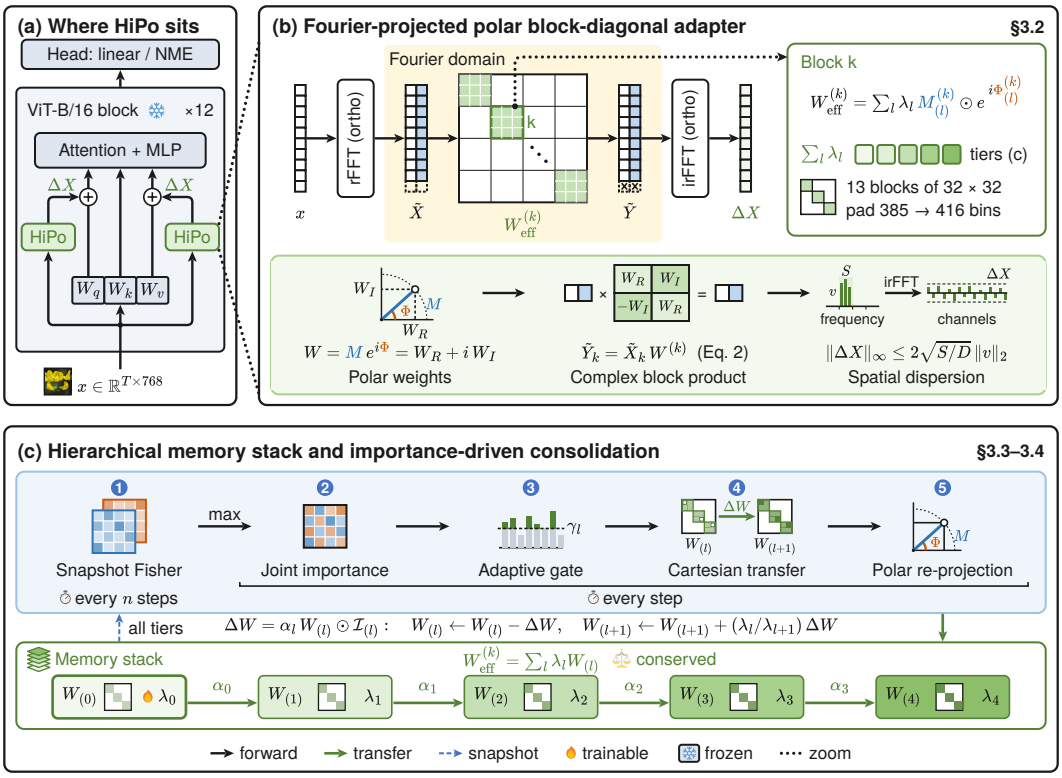


Figure 2: **Overview of HiPo.** Input features undergo an orthonormal real FFT and are processed by polar-parameterized complex blocks (§3.2). A fixed-depth stack combines a trainable working tier with deeper tiers (§3.3); periodic Snapshot Fisher estimates guide continuous importance-gated Cartesian transfers between tiers (§3.4).

to distribute block-local updates across output coordinates. We characterize this spatial dispersion through a per-coordinate displacement bound (Appendix D.4), providing theoretical motivation for our long-sequence experiments.

Fixed Fourier Mixing. Let $X \in \mathbb{R}^{B\times T\times D}$ denote the input token representations. We apply a one-dimensional real Fast Fourier Transform (rFFT) along the feature dimension D as a parameter-free dense mixing operation, rather than to extract physical periodicities from the unordered feature channels. The orthonormal transform $\mathcal{F}_{\text{ortho}} : \mathbb{R}^D \to \mathbb{C}^F$, where $F = \lfloor D/2 \rfloor + 1$, has computational complexity $\mathcal{O}(D \log D)$. By mixing features globally, the transform allows localized updates in the Fourier domain to influence multiple spatial coordinates after reconstruction. Appendix D.4 characterizes this dispersion through a per-coordinate displacement bound that accounts for Hermitian symmetry and orthonormal scaling.

Polar Optimization and Initialization. In the complex Fourier space, we parameterize adapter weights using magnitude M and phase Φ as $W = M \odot e^{i\Phi}$, allowing these components to be optimized separately. Since phase gradients are proportional to magnitude, initializing M at zero would suppress phase learning. We therefore initialize $M \sim \mathcal{U}(0, \epsilon)$ with $\epsilon = 0.02$ and $\Phi \sim \mathcal{U}(-\pi, \pi)$. The complex weights are computed through their Cartesian components:

$$W = M \odot \cos(\Phi) + i(M \odot \sin(\Phi)) = W_R + iW_I. \tag{1}$$

This formulation avoids division by magnitude, while nonzero initialization supports phase-gradient learning (Appendix C.1).

Block-Diagonal Complex Transformation. To achieve parameter-efficient adaptation, we organize W into block-diagonal matrices while retaining the global feature mixing provided by the Fourier transform. The F complex channels are zero-padded to $\tilde{F}$ and partitioned into blocks of size $S \times S$. Each block k independently applies a complex linear transformation $\tilde{Y}^{(k)} = \tilde{X}^{(k)} W^{(k)}$, computed

as:

$$\begin{aligned}\Re(\tilde{Y}^{(k)}) &= \Re(\tilde{X}^{(k)})W_R^{(k)} - \Im(\tilde{X}^{(k)})W_I^{(k)},\\ \Im(\tilde{Y}^{(k)}) &= \Re(\tilde{X}^{(k)})W_I^{(k)} + \Im(\tilde{X}^{(k)})W_R^{(k)}.\end{aligned} \tag{2}$$

After the block transformation, we remove the padding and retain $F = 385$ complex bins for $D = 768$. The transformed features are mapped back to the real-valued feature space using an inverse real FFT:

$$\Delta X = \mathrm{irFFT}(\tilde{Y}_{:F}; n = D, \mathrm{norm} = \mathrm{ortho}). \tag{3}$$

The resulting residual ΔX is added to the query and value outputs of each adapted qkv projection, leaving the key output unchanged. The inverse transform handles Hermitian reconstruction and restores the original feature dimension without concatenating real and imaginary components.

3.3 Hierarchical Stacked Memory Architecture

Inspired by the different learning timescales of Complementary Learning Systems (29; 20), we organize the adapters into a fixed-depth hierarchy of L tiers. The shallowest tier ($l = 0$) serves as the plastic Working Memory (WM), while deeper tiers ($l > 0$) form the Long-Term Memory (LTM). For each frequency block k, the output combines contributions from all tiers through learnable superposition weights:

$$\tilde{Y}^{(k)} = \tilde{X}^{(k)} \sum_{l=0}^{L-1} \lambda_l \left(M_{(l)}^{(k)} \odot e^{i\Phi_{(l)}^{(k)}} \right), \tag{4}$$

where $\lambda_l \in \mathbb{R}$ determines the contribution of tier l. These learnable weights allow the relative influence of each memory tier to adapt throughout training.

Tier-specific update dynamics. During task $\mathcal{T}_t$, direct task-loss gradients update only the working-memory adapter parameters, while the superposition weights λ_l remain trainable:

$$\begin{aligned}M_{(0),\tau+1}^{(k)} &\leftarrow \mathrm{Optimizer}\left(M_{(0),\tau}^{(k)}, \nabla_M \mathcal{L}_{CE}\right),\\ \Phi_{(0),\tau+1}^{(k)} &\leftarrow \mathrm{Optimizer}\left(\Phi_{(0),\tau}^{(k)}, \nabla_\Phi \mathcal{L}_{CE}\right).\end{aligned} \tag{5}$$

The long-term memory tiers receive knowledge through gradual inter-tier transfers rather than direct task-loss updates. Their parameters are temporarily enabled for gradient computation during Snapshot Fisher estimation. To establish progressively slower consolidation across the hierarchy, the transfer rate from tier l to $l + 1$ decreases exponentially with depth:

$$\alpha_l = \frac{1}{f_0 \cdot \gamma^l}, \tag{6}$$

where f_0 controls the base consolidation rate and $\gamma > 1$ determines the rate of decay.

Sequential Cartesian transfer. Knowledge consolidation operates continuously during training. Snapshot Fisher importance scores are refreshed every f_{freq} optimization steps, while importance-gated transfers are performed at every step using the latest estimates. Transfers proceed sequentially from shallow to deeper tiers, with each pair $(l, l + 1)$ updated in place before processing the next pair.

Let $W_{(l)}^{[l]}$ denote the complex weight of tier l immediately before its transfer step. The importance-weighted transfer is defined as

$$\begin{aligned}\Delta W_{(l)} &= \alpha_l W_{(l)}^{[l]} \odot \mathcal{I}_{(l)},\\ W_{(l)}^{[l+1]} &= W_{(l)}^{[l]} - \Delta W_{(l)},\\ W_{(l+1)}^{[l+1]} &= W_{(l+1)}^{[l]} + r_l \Delta W_{(l)}, \qquad r_l = \frac{\lambda_l}{\lambda_{l+1}},\end{aligned} \tag{7}$$

where $\mathcal{I}_{(l)} \in [0, 1]$ is the Fisher-derived importance mask, and all other tiers remain unchanged during the pairwise update. The in-place procedure allows intermediate tiers to receive and subsequently transfer knowledge within the same cascade. The scaling factor r_l compensates for differences in superposition weights, preserving the effective adapter weight $\sum_l \lambda_l W_{(l)}$ during each transfer when λ_l remain fixed and the numerical denominator clamp is inactive.

Following each transfer, the updated Cartesian complex weights $\widehat{W}_{(l)}^{(k)}$ are converted back to polar coordinates:

$$\begin{aligned} M_{(l)}^{(k)} &\leftarrow \sqrt{\Re\big(\widehat{W}_{(l)}^{(k)}\big)^2 + \Im\big(\widehat{W}_{(l)}^{(k)}\big)^2}, \\ \Phi_{(l)}^{(k)} &\leftarrow \operatorname{atan2}\Big(\Im\big(\widehat{W}_{(l)}^{(k)}\big), \Re\big(\widehat{W}_{(l)}^{(k)}\big)\Big). \end{aligned} \tag{8}$$

This process redistributes complex adapter weights across memory tiers while maintaining their polar parameterization for subsequent optimization.

Hierarchical knowledge retention. The hierarchy separates active adaptation in working memory from the accumulated contributions of deeper tiers. For each frequency block k, define the aggregate long-term memory contribution as $C^{(k)} = \sum_{l=1}^{L-1} \lambda_l W_{(l)}^{(k)}$. The effective transformation can then be expressed as

$$\tilde{Y}^{(k)} = \lambda_0 \tilde{X}^{(k)} W_{(0)}^{(k)} + \tilde{X}^{(k)} C^{(k)}. \tag{9}$$

This decomposition illustrates how the current task is learned through the working tier while the deeper tiers contribute previously consolidated weights. The relative contribution of long-term memory is governed by the learned superposition weights, which may evolve throughout training. Appendix D.3 further analyzes the role of the deep-tier contribution under specified conditions.

3.4 Importance-driven Knowledge Consolidation

To guide knowledge transfer across memory tiers, we introduce a parameter-specific importance mask $\mathcal{I}_{(l)}$ that controls the fractional transfer ΔW (Eq. 7). Parameters with higher estimated importance receive stronger transfer weights, allowing influential updates to be preferentially consolidated into deeper memory tiers.

Disentangled Polar Snapshot Fisher. Building on Fisher-information-based approaches to continual learning (17), we estimate parameter importance using an empirical Snapshot Fisher computed periodically over sampled batches $\mathcal{S}_\tau \subset \mathcal{D}_t$. Importance is estimated separately for magnitude and phase parameters through squared loss gradients:

$$\begin{aligned} F_M &= \mathbb{E}_{x\sim\mathcal{S}_\tau}\left[\sum_{l=0}^{L-1}\left(\nabla_{M_{(l)}}\mathcal{L}_{CE}\right)^2\right], \\ F_\Phi &= \mathbb{E}_{x\sim\mathcal{S}_\tau}\left[\sum_{l=0}^{L-1}\left(\nabla_{\Phi_{(l)}}\mathcal{L}_{CE}\right)^2\right]. \end{aligned} \tag{10}$$

The squared gradients are accumulated across all tiers into shared importance buffers, which provide consistent importance estimates for each adjacent tier pair $(l, l+1)$.

Adaptive Importance Thresholding. To obtain comparable importance scores, we standardize the magnitude and phase Fisher estimates using their respective statistics and apply a sigmoid transformation:

$$\begin{aligned} P_M &= \operatorname{Sigmoid}\left(\frac{F_M - \mu_{F_M}}{\sigma_{F_M}}\right), \\ P_\Phi &= \operatorname{Sigmoid}\left(\frac{F_\Phi - \mu_{F_\Phi}}{\sigma_{F_\Phi}}\right). \end{aligned} \tag{11}$$

We then combine both components through an element-wise maximum, $P_{\text{joint}} = \max(P_M, P_\Phi)$, allowing importance in either magnitude or phase to contribute to consolidation.

Each tier boundary uses an adaptive threshold γ_l to regulate the proportion of parameters selected for transfer. A proportional feedback controller adjusts this threshold toward a target gate pass rate ρ_l:

$$\gamma_l \leftarrow \Pi_{[0.05,\,0.99]}\left(\gamma_l + \eta\left[\mathbb{E}\left[\mathbb{I}(P_{\text{joint}} > \gamma_l)\right] - \rho_l\right]\right), \tag{12}$$

where η is the controller learning rate, $\mathbb{I}(\cdot)$ is the indicator function, and Π projects the threshold onto the specified interval. The gate pass rate is computed from query-branch importance scores averaged across blocks, and the resulting threshold is shared with the value branch. This feedback mechanism adjusts the selection threshold as importance distributions evolve, approximately tracking the target rate under the fixed-step implementation. Appendix D.6 provides the corresponding stability analysis.

Continuous Importance-Gated Transfer. The importance mask for each block k is defined as

$$\mathcal{I}_{(l)}^{(k)} = \mathrm{ReLU}\left(P_{\mathrm{joint}}^{(k)} - \gamma_l\right). \tag{13}$$

Snapshot Fisher estimates are periodically refreshed and remain fixed between updates, while importance-gated Cartesian transfers occur at every optimization step. This enables gradual redistribution of adapter weights from working memory to deeper tiers as new learning updates replenish the active tier. The resulting magnitude dynamics are illustrated in Figure 6. Furthermore, the polar parameterization introduces magnitude-proportional phase-gradient sensitivity (Appendix C.1), while component ablations assess the contributions of the adaptation and consolidation mechanisms to knowledge retention.

4 Experiments

4.1 Experimental setup

Datasets and Evaluation. We evaluate HiPo under the strictly exemplar-free CIL paradigm (zero buffered data) across four benchmarks: CIFAR-100 (19), CUB-200 (40), ImageNet-R (12), and ImageNet-A (13). ImageNet-R and ImageNet-A probe adaptation under two types of distribution shift. Both were originally released as evaluation image collections; their training/evaluation partitions in CIL are protocol-specific, and ImageNet-A has no official training split (12; 13). Following the experimental setup of (44), CIFAR-100, CUB-200, and ImageNet-A are uniformly partitioned into $N = 10$ sequential tasks. To examine task-sequence fragmentation, ImageNet-R is evaluated with $N \in \{5, 10, 20\}$ tasks. Performance is measured using final Average Accuracy (Acc) and Average Anytime Accuracy (AAA), reported as the mean and standard deviation (mean $\pm$ std) over five independent runs.

Implementation Details. We leverage a permanently frozen pre-trained Vision Transformer (ViT-B/16) backbone (8) to preserve generalized feature extraction. The full model uses the same $L = 5$ stack and fixed configuration for all datasets, sequence lengths and evaluation heads; depth L and the initial threshold schedule γ_l are the two principal architectural sensitivity variables. Comprehensive implementation details, dataset statistics, and hardware specifications are provided in Appendix B.

4.2 Baselines and prediction strategies

We compare prompt-based approaches (L2P, DualPrompt, CODA-Prompt, HiDe-Prompt), orthogonal/adaptive methods (O-LoRA, InfLoRA, SD-LoRA, BiLoRA), and RanPAC (44; 43; 36; 41; 42; 23; 45; 48; 31). All reported experiments were run in a single PILOT-based evaluation codebase (37) using the same frozen `timm` ViT-B/16, data splits, class-order seeds $\{14, 15, 16, 17, 18\}$ and zero-buffer condition on NVIDIA P100 hardware. Methods ported into PILOT retain their published adapter architectures, losses, optimizers, learning rates and epoch schedules. The baselines share data and evaluation infrastructure but retain their published training configurations; the parameter-matched ablations provide more controlled comparisons of individual components.

We report (i) an expanding parametric linear classification head (CNN) and (ii) the established nearest-mean-of-exemplars (NME) classifier, in which class prototypes are extracted from each newly observed training set at the end of that task and stored without images. Normalized feature distances determine predictions. Reporting both heads distinguishes adapter representation quality from prediction-head effects: the CNN head performs better in some settings, including CIFAR-100 and short ImageNet-R sequences, while matched-NME results directly compare prototype-based evaluation.

4.3 Main results and analysis

Table 1 summarizes the Average Anytime Accuracy (AAA) and final Average Accuracy (Acc) across all CIL benchmarks.

Task fragmentation and capacity. HiPo maintains 79.21% and 79.00% final NME accuracy on ImageNet-R with 10 and 20 tasks, respectively, a decline of just 0.21 points. In comparison, matched SD-LoRA (NME), RanPAC (full) and BiLoRA decline by 1.38, 2.63 and 4.04 points. HiPo's

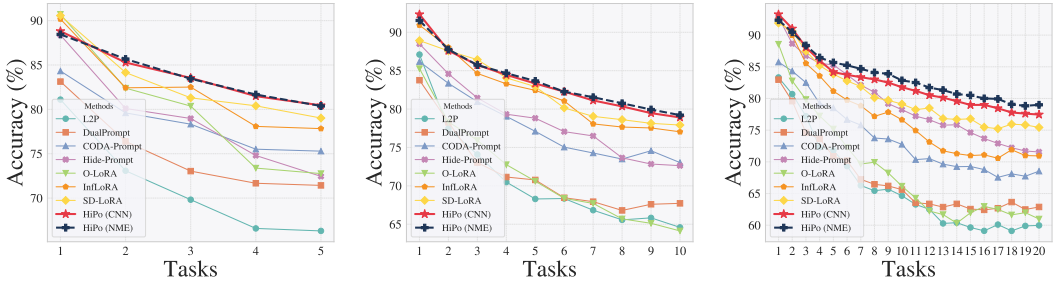
Table 1: Average Anytime Accuracy (AAA) and final Average Accuracy (Acc) across CIL benchmarks, reported as mean ± standard deviation over five runs. CNN denotes the parametric classifier and NME the standard nearest-mean classifier. Best results are in bold, and second-best results are underlined.

Method	IN-R (5T)		IN-R (10T)		IN-R (20T)		IN-A (10T)		CIFAR-100 (10T)		CUB-200 (10T)	
	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc
L2P (44)	71.54 ±0.64	66.30 ±0.35	70.62 ±0.23	64.59 ±0.48	67.35 ±0.53	59.97 ±0.31	50.61 ±0.21	41.17 ±0.39	87.58 ±0.52	82.76 ±0.98	69.30 ±0.39	56.48 ±0.29
DualPrompt (43)	73.28 ±0.41	71.43 ±0.46	71.23 ±0.74	67.72 ±0.54	68.61 ±0.74	62.88 ±0.80	54.86 ±0.53	43.07 ±0.29	90.57 ±0.23	83.88 ±0.48	78.57 ±0.54	67.26 ±0.49
CODA-Prompt (36)	78.68 ±0.23	75.29 ±0.52	77.11 ±0.71	73.03 ±0.63	73.71 ±0.55	68.53 ±0.61	56.30 ±0.84	45.94 ±0.49	91.37 ±0.27	86.82 ±0.69	82.36 ±0.40	72.41 ±0.63
HiDe-Prompt (41)	77.42 ±0.21	72.45 ±0.31	76.84 ±0.35	72.63 ±0.61	75.88 ±0.25	71.56 ±0.66	55.72 ±0.44	44.67 ±0.82	91.17 ±0.34	86.53 ±0.31	81.71 ±0.51	74.15 ±0.72
O-LoRA (42)	76.71 ±1.81	72.76 ±1.52	71.30 ±1.02	64.13 ±1.86	68.04 ±0.36	60.98 ±0.98	54.17 ±0.81	44.46 ±1.48	89.96 ±1.43	82.51 ±1.52	79.66 ±1.57	70.12 ±0.40
InfLoRA (23)	81.63 ±0.27	77.83 ±0.33	80.24 ±0.36	77.04 ±0.68	76.78 ±0.86	70.94 ±0.90	61.63 ±0.19	51.02 ±0.26	91.44 ±0.46	86.09 ±0.28	81.43 ±0.46	71.13 ±0.67
SD-LoRA (45)	82.85 ±0.40	79.02 ±0.28	81.05 ±0.42	77.92 ±0.22	80.62 ±0.50	75.43 ±0.57	63.24 ±0.20	55.35 ±0.47	91.71 ±0.33	87.28 ±0.23	85.15 ±0.55	77.38 ±0.35
SD-LoRA (NME) (45)	83.31 ±0.67	78.63 ±0.71	81.29 ±0.47	78.02 ±0.23	81.72 ±0.48	76.64 ±0.83	68.66 ±0.26	63.98 ±0.43	92.07 ±0.47	87.73 ±0.12	85.63 ±0.68	80.50 ±0.33
BiLoRA (48)	83.02 ±0.47	77.55 ±0.37	82.48 ±0.29	77.50 ±0.73	80.38 ±0.61	73.46 ±0.26	67.83 ±0.64	56.45 ±0.22	92.65 ±0.48	88.93 ±0.67	**86.83 ±0.33**	81.56 ±0.65
RanPAC (full) (31)	82.95 ±0.77	79.65 ±0.55	82.61 ±0.59	78.14 ±0.32	79.77 ±0.48	75.51 ±0.52	70.17 ±0.82	58.33 ±0.73	**93.11 ±0.87**	**89.68 ±0.80**	86.64 ±0.52	**82.08 ±0.61**
HiPo (CNN)	83.87 ±0.25	**80.45 ±0.37**	83.56 ±0.41	78.91 ±0.14	82.37 ±0.44	77.46 ±0.36	65.58 ±0.13	56.41 ±0.36	91.93 ±0.28	88.54 ±0.05	86.78 ±0.32	76.15 ±0.48
HiPo (NME)	**83.90 ±0.25**	80.38 ±0.19	**83.75 ±0.40**	**79.21 ±0.45**	**83.37 ±0.32**	**79.00 ±0.37**	**71.37 ±0.88**	**64.39 ±0.52**	91.57 ±0.18	88.28 ±0.07	86.45 ±0.32	81.34 ±0.22

advantage over the strongest of these baselines widens from 0.73 points at 5 tasks to 2.36 points at 20 tasks. Against BiLoRA at 20 tasks, it achieves 5.54 points higher final accuracy with approximately 0.78M versus 1.59M adapter parameters. Together with the feature-drift measurements, these results support a capacity-preserving explanation for its robust long-sequence behavior; the projection analysis separately bounds per-update displacement.

Distribution shift and classifier-dependent performance. On ImageNet-A, HiPo reaches 64.39% NME accuracy, compared with 63.98% for SD-LoRA under the same NME protocol, 58.33% for full RanPAC and 56.45% for BiLoRA. The basis/coordinate ablation shows a larger combined gain over spatial-Cartesian adaptation on ImageNet-A than on ImageNet-R, while the isolated polar-coordinate effect depends on the prediction head. On CIFAR-100 and CUB-200, RanPAC attains 89.68% and 82.08% final accuracy, compared with HiPo NME's 88.28% and 81.34%; BiLoRA obtains 88.93% and 81.56%. These comparisons highlight the complementary importance of strong prediction heads on in-distribution benchmarks and contextualize HiPo's particular strength in adapting representations under shift.

Feature geometry. Figure 4 visualizes the evolution of HiPo's class clusters across tasks. NME results complement this qualitative view by assessing feature separability without a separately trained parametric head. Direct measurements further show similar mean old-feature drift at the final task for ImageNet-R 10T (0.370) and 20T (0.361), consistent with the stable long-sequence performance. The t-SNE visualization is descriptive, and Appendix F specifies the measurement protocol and limitations.



(a) ImageNet-R (5 tasks) (b) ImageNet-R (10 tasks) (c) ImageNet-R (20 tasks)

Figure 3: Task-wise ImageNet-R accuracy across 5, 10 and 20 tasks. HiPo maintains strong retention across increasing sequence fragmentation; Table 1 provides matched-protocol final-accuracy comparisons.

4.4 Ablation study

We evaluate each component on ImageNet-R and ImageNet-A using parameter-matched variants. A factorial comparison additionally separates the effects of the projection basis and the parameter coordinate system.

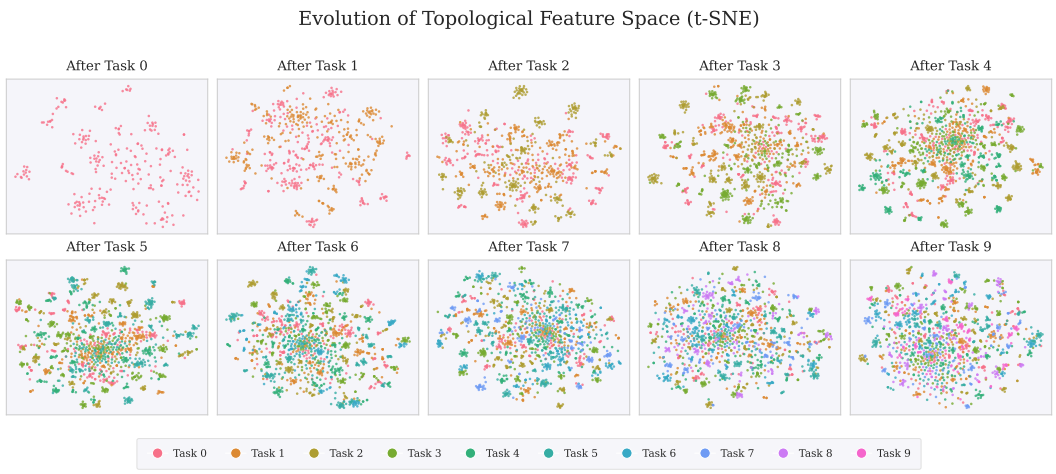


Figure 4: **Qualitative class-feature visualization** with 2D t-SNE projections across a 10-task sequence.

Table 2: NME ablation results on ImageNet-R (5, 10 and 20 tasks) and ImageNet-A (10 tasks), reported as mean ± standard deviation across five runs. Best results are in bold, and second-best results are underlined.

Ablation Variant (NME)	**IN-R (5 tasks)**		**IN-R (10 tasks)**		**IN-R (20 tasks)**		**IN-A (10 tasks)**	
	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc
w/o Orthogonal Proj. (Spatial-Polar)	78.05 ±0.43	71.58 ±0.43	79.19 ±0.89	72.93 ±0.60	78.64 ±0.66	71.45 ±0.31	59.45 ±0.58	49.78 ±0.55
w/o Polar Disentanglement (Freq.-Cart.)	83.02 ±0.37	79.89 ±0.27	83.04 ±0.59	**79.51 ±0.55**	83.13 ±0.61	78.91 ±0.89	71.38 ±0.52	63.87 ±0.58
w/o Hierarchical Consolidation	83.34 ±0.42	80.36 ±0.31	82.78 ±0.38	78.66 ±0.29	83.21 ±0.49	**79.11 ±0.33**	71.01 ±1.16	63.35 ±0.28
w/o Cartesian-Invariant Transfer	82.71 ±0.16	79.34 ±0.77	82.52 ±0.24	78.90 ±0.12	82.87 ±0.50	78.95 ±0.53	**71.59 ±0.74**	62.87 ±0.75
HiPo (Full)	**83.90 ±0.25**	**80.38 ±0.19**	**83.75 ±0.40**	79.21 ±0.45	**83.37 ±0.32**	79.00 ±0.37	71.37 ±0.88	**64.39 ±0.52**

4.4.1 Projection, coordinates and consolidation

Factorial interpretation. The ablations jointly vary basis and coordinate system. On ImageNet-A (NME/CNN), Spatial-Cartesian obtains 63.15/54.23, Spatial-Polar 49.78/49.45, Frequency-Cartesian 63.87/56.55, and Frequency-Polar (HiPo) 64.39/56.41. Frequency mixing improves the Cartesian adapter by 0.72 NME / 2.32 CNN points; polar coordinates add 0.52 NME / −0.14 CNN points conditional on the frequency basis. Comparing the full model with spatial-Cartesian yields 1.24 NME / 2.18 CNN points. The large Spatial-Polar deficit (14.61 NME points relative to full HiPo) shows an interaction between the coordinate system and basis, not that an orthogonal basis is a strict mathematical prerequisite for every adapter. On ImageNet-R, the 0.08–0.66-point gains over Spatial-Cartesian are small relative to run-to-run differences.

Retention versus acquisition. Table 3 reports immediate learning accuracy and subsequent forgetting as complementary measures of acquisition and retention. HiPo achieves the lowest forgetting across all four settings, revealing a retention advantage that can be obscured when ablations exhibit higher immediate learning accuracy. For example, Frequency-Cartesian achieves 79.51 versus HiPo's 79.21 final NME accuracy on ImageNet-R 10T, while removing consolidation yields 79.11 versus 79.00 on ImageNet-R 20T (Table 2). These results distinguish HiPo's strong retention from gains in initial acquisition and help localize the contributions of its components.

Continuous transfer. The hierarchy contributes 1.04 points on ImageNet-A NME (64.39 versus 63.35 without consolidation); Cartesian transfer also improves over direct polar transfer (64.39 versus 62.87). Figure 6 illustrates the gradual working-memory magnitude changes observed with continuous importance-gated transfer, alongside small phase gradients. The effects are clearest under ImageNet-A shift, while ImageNet-R gains are smaller. Learned deep-tier superposition coefficients decrease over training, and the depth/retention ablations show a non-monotonic pattern consistent with a capacity–regularization tradeoff (Appendix E.2).

Table 3: Immediate learning accuracy (Learn) and average forgetting of previously learned tasks (Forget), in percentage points. Each cell is Learn / Forget; lower Forget means better retention. These averages are computed over different task sets, so Learn − Forget need not equal final Acc. Best results are in bold, and second-best results are underlined.

Variant	IN-R 5T	IN-R 10T	IN-R 20T	IN-A 10T
w/o Orthogonal Proj.	81.38 / 12.25	82.53 / 10.67	83.46 / 12.64	64.92 / 16.82
w/o Polar Disentang.	**87.60** / 9.63	**88.66** / 10.17	**89.11** / 10.74	**78.21** / 15.93
w/o Hierarchical Consol.	87.06 / 8.38	87.89 / 10.25	88.38 / 9.76	76.90 / 15.05
w/o Cartesian-Inv. Trans.	87.27 / 9.91	87.63 / 9.70	88.45 / 10.00	76.20 / 14.81
HiPo (full)	86.98 / **8.25**	87.87 / **9.62**	88.08 / **9.55**	76.59 / **13.56**

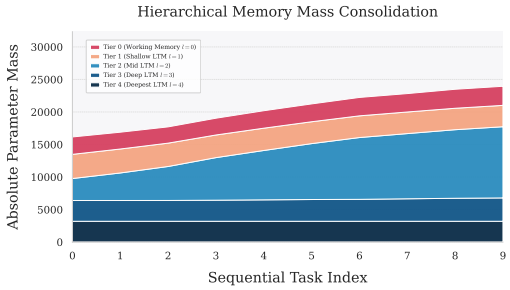


Figure 5: **Macroscopic consolidation** of HiPo visualized with the distribution of absolute parameter mass (L_2 norm).

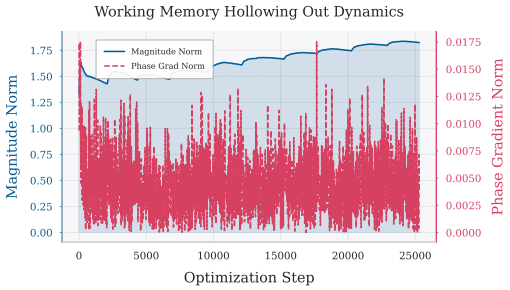


Figure 6: **Working-memory dynamics.** Magnitude norm (blue) and phase-gradient magnitude (red dashed) evolve under continuous importance-gated transfer, with periodic Snapshot Fisher updates regulating consolidation.

4.4.2 Sensitivity to stack depth and retention thresholds

Table 6 varies stack depth L and the initial threshold schedule γ_l while fixing other settings. On ImageNet-A NME, $L = 1, 3, 5, 7$ gives 62.99, 64.07, 64.39 and 62.95, respectively, with the 3- and 5-tier results overlapping within reported variation. The reference initial threshold schedule reaches 64.39, compared with 62.91 and 62.79 for its permissive and restrictive alternatives. These results identify a useful intermediate operating regime and characterize the hierarchy's capacity–regularization tradeoff without relying on a literal biological interpretation.

5 Conclusion

We presented HiPo for exemplar-free class-incremental learning, combining Fourier-projected polar block-diagonal adapters, a fixed-depth hierarchical memory stack, and Snapshot Fisher-guided Cartesian consolidation. The Fourier transform disperses localized adapter updates across spatial coordinates; the memory hierarchy separates task optimization from gradual storage without expanding the adapter stack with each task. Adaptive importance gating enables continuous transfers that preserve the effective adapter weight when the numerical clamp is inactive. Across the evaluated benchmarks, HiPo retains accuracy particularly well as ImageNet-R task sequences become more fragmented and performs competitively on ImageNet-A. Controlled ablations show that the Fourier basis is especially important in combination with polar adaptation, while the isolated polar and consolidation gains are smaller and depend on the setting. The theoretical results characterize update geometry and transfer algebra, not guaranteed retention of past-task predictions. Future work will evaluate alternative unitary bases, compare feature drift directly with subspace baselines, and study larger backbones and task distributions.

References

[1] Amari, S.-I. (1998) Natural gradient works efficiently in learning. *Neural Computation* **10**(2):251–276.

[2] Arjovsky, M., Shah, A. & Bengio, Y. (2016) Unitary evolution recurrent neural networks. In *International Conference on Machine Learning (ICML)*.

[3] Behrouz, A., Razaviyayn, M., Zhong, P. & Mirrokni, V. (2025) Nested learning: The illusion of deep learning architectures. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[4] Chaudhry, A., Dokania, P.K., Ajanthan, T. & Torr, P.H.S. (2018) Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*.

[5] Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J. & Luo, P. (2022) AdaptFormer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[6] Dettmers, T., Pagnoni, A., Holtzman, A. & Zettlemoyer, L. (2023) QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[7] Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H.-T., Chen, J., Liu, Y., Tang, J., Li, J. & Sun, M. (2023) Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* **5**(3):220–235.

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. (2021) An image is worth 16×16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

[9] Farajtabar, M., Azizan, N., Mott, A. & Li, A. (2020) Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[10] Guibas, J., Mardani, M., Li, Z., Tao, A., Anandkumar, A. & Catanzaro, B. (2022) Efficient token mixing for transformers via adaptive Fourier neural operators. In *International Conference on Learning Representations (ICLR)*.

[11] He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T. & Neubig, G. (2022) Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations (ICLR)*.

[12] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, A., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J. & Gilmer, J. (2021a) The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[13] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J. & Song, D. (2021b) Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[14] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M. & Gelly, S. (2019) Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*.

[15] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. & Chen, W. (2022) LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

[16] Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B. & Lim, S.-N. (2022) Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*.

[17] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D. & Hadsell, R. (2017) Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* **114**(13):3521–3526.

[18] Kopiczko, D.J., Blankevoort, T. & Asano, Y.M. (2024) VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations (ICLR)*.

[19] Krizhevsky, A. (2009) Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[20] Kumaran, D., Hassabis, D. & McClelland, J.L. (2016) What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences* **20**(7):512–534.

[21] Lee-Thorp, J., Ainslie, J., Eckstein, I. & Ontañón, S. (2022) FNet: Mixing tokens with Fourier transforms. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[22] Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2021) Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*.

[23] Liang, Y.-S. & Li, W.-J. (2024) InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[24] Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M. & Raffel, C. (2022) Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[25] Liu, R., Diao, B., Huang, L., An, Z., An, Z. & Xu, Y. (2024) Continual learning in the frequency domain. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[26] Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C.F., Cheng, K.-T. & Chen, M.-H. (2024) DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning (ICML)*.

[27] Lopez-Paz, D. & Ranzato, M.A. (2017) Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[28] Martens, J. (2020) New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research* **21**(146):1–76.

[29] McClelland, J.L., McNaughton, B.L. & O'Reilly, R.C. (1995) Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review* **102**(3):419–457.

[30] McCloskey, M. & Cohen, N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation* **24**:109–165.

[31] McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E. & van den Hengel, A. (2023) RanPAC: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[32] Rebuffi, S.-A., Kolesnikov, A., Sperl, G. & Lampert, C.H. (2017) iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[33] Ritter, H., Botev, A. & Barber, D. (2018) Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[34] Saha, G., Garg, I. & Roy, K. (2021) Gradient projection memory for continual learning. In *International Conference on Learning Representations (ICLR)*.

[35] Schwarz, J., Czarnecki, W.M., Luketina, J., Grabska-Barwinska, A., Teh, Y.W., Pascanu, R. & Hadsell, R. (2018) Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning (ICML)*.

[36] Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R. & Kira, Z. (2023) CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Sun, H.-L., Zhou, D.-W., Zhan, D.-C. & Ye, H.-J. (2025) PILOT: A pre-trained model-based continual learning toolbox. *Science China Information Sciences* **68**(4):147101.

[38] Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y. & Pal, C.J. (2018) Deep complex networks. In *International Conference on Learning Representations (ICLR)*.

[39] van de Ven, G.M., Tuytelaars, T. & Tolias, A.S. (2022) Three types of incremental learning. *Nature Machine Intelligence* **4**(12):1185–1197.

[40] Wah, C., Branson, S., Welinder, P., Perona, P. & Belongie, S. (2011) The Caltech-UCSD Birds-200-2011 dataset. Computation & Neural Systems Technical Report, CNS-TR-2011-001.

[41] Wang, L., Xie, J., Zhang, X., Huang, M., Su, H. & Zhu, J. (2023) Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[42] Wang, X., Chen, T., Ge, Q., Xia, H., Bao, R., Zheng, R., Zhang, Q., Gui, T. & Huang, X. (2023) Orthogonal subspace learning for language model continual learning. In *Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[43] Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J. & Pfister, T. (2022a) DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision (ECCV)*.

[44] Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J. & Pfister, T. (2022b) Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[45] Wu, Y., Piao, H., Huang, L.-K., Wang, R., Li, W., Pfister, H., Meng, D., Ma, K. & Wei, Y. (2025) SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *International Conference on Learning Representations (ICLR)*.

[46] Zenke, F., Poole, B. & Ganguli, S. (2017) Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*.

[47] Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W. & Zhao, T. (2023) Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*.

[48] Zhu, H., Zhang, Y., Dong, J. & Koniusz, P. (2025) BiLoRA: Almost-orthogonal parameter spaces for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Funding and competing-interest disclosures

Funding: This work was supported by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under the project "Research and Development of a Personalized Machine Learning System for Early Alzheimer's Disease Diagnosis from Adaptive Multimodal Biomarkers" (Grant No. 102.05-2025.54).

Competing interests: The authors declare no competing interests.

A Algorithmic framework

This section formalizes the HiPo (Hierarchical Polar Adapters) framework and distinguishes Snapshot Fisher refreshes from the continuously applied transfer. We present the exact sequence of operations for the Class-Incremental Learning (CIL) loop (Algorithm 1) and our Importance-Driven Trickle-Down Consolidation mechanism (Algorithm 2). Furthermore, we document critical numerical and indexing heuristics utilized in the implementation to improve numerical stability and to clarify the separation of the active and stored parameter tiers.

A.1 Notation and system variables

To ensure exact correspondence with the main paper, we maintain the following notation:

- $\mathcal{T}_t$: The current active task with dataset $\mathcal{D}_t$ and labels bounded by $[|\mathcal{Y}_{1:t-1}|, |\mathcal{Y}_{1:t}| - 1]$.
- Θ_{fc}: The incrementally expanding linear classification head.
- θ: The set of HiPo adapter parameters, specifically partitioned into the highly plastic Working Memory ($l = 0$) and frozen Long-Term Memory (LTM) tiers ($l \in \{1, \dots, L-1\}$).
- $M_{(l)}, \Phi_{(l)}$: The magnitude and phase parameters for tier l.

- λ_l: Dynamically learnable superposition scaling weights for tier l.
- F_M, F_Φ: Exponential moving average (EMA) Empirical Snapshot Fisher buffers, accumulated over all tiers and shared by all tier pairs.
- γ_l, ρ_l: The adaptive tracking threshold and the target parameter retention rate for tier l, respectively.

A.2 Formal algorithms

Below, Algorithm 1 outlines the macroscopic training loop across episodic tasks. Algorithm 2 details a periodic Snapshot Fisher refresh followed by transfer at each step. The clamped transfer is not exactly invariant.

Algorithm 1 Incremental Training Pipeline (HiPo CIL Loop)

Require: Task sequence $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, Pre-trained frozen backbone ϕ, Stack depth L.
Require: Snapshot frequency f_{freq}, Optimizer (SGD with momentum), Learning rate scheduler.
Initialize: Classification head $\Theta_{fc} \leftarrow \emptyset$. Global step $\tau \leftarrow 0$.
Initialize: HiPo Adapters: $M_{(0)} \sim \mathcal{U}(0, 0.02)$, $\Phi_{(0)} \sim \mathcal{U}(-\pi, \pi)$. LTM tiers ($l > 0$) locked and initialized to 0.
for task $t = 1$ **to** N **do**
  Expand Θ_{fc} to accommodate novel classes of $\mathcal{T}_t$.
  $c_{\text{prev}} \leftarrow |\mathcal{Y}_{1:t-1}|$ {Classes seen before this task}
  Enable gradient updates for working-memory magnitude and phase.
  for each epoch **do**
    for each mini-batch $(X, y) \in \mathcal{D}_t$ **do**
      $\tau \leftarrow \tau + 1$
      1. Continuous transfer, with periodic Snapshot Fisher refresh
      Execute **Algorithm 2** with refresh flag $s = [\tau \bmod f_{\text{freq}} = 0]$
      2. Standard Optimization Step
      Zero gradients for Optimizer.
      Extract features via backbone ϕ and Working Memory $M_{(0)}, \Phi_{(0)}$.
      Logits $z \leftarrow \text{Linear}_{\Theta_{fc}}(\text{Features})$
      3. CIL Masking (Task Isolation)
      Mask active logits: $z_{\text{new}} \leftarrow z[:, c_{\text{prev}} :]$
      Re-index active targets: $y_{\text{active}} \leftarrow y - c_{\text{prev}}$
      $\mathcal{L}_{CE} \leftarrow \text{CrossEntropy}(z_{\text{new}}, y_{\text{active}})$
      Compute gradients of $\mathcal{L}_{CE}$
      Update working-memory parameters and current classification head
    end for
    Update learning-rate schedule
  end for
end for

A.3 Critical implementation nuances

For reproducibility, we specify the implementation conventions used for Cartesian reprojection, Fisher refreshes, and continuous transfer:

1. Handling the origin singularity ($M \to 0$). The Cartesian forward mapping $W = M \odot (\cos\Phi + i \sin\Phi)$ avoids differentiation through an inverse-angle mapping during ordinary training. The phase gradient is magnitude-proportional under bounded Cartesian gradients. Reprojection after transfer uses atan2; phase at exactly zero magnitude is not unique, but has no effect on the forward weight.

2. Continuous transfer and Fisher snapshots. Snapshot gradients and Fisher estimates are refreshed every $f_{\text{freq}} = 100$ steps; the last estimate is reused for the transfer performed at every step, which begins once the first snapshot is available. Thus, the observed smooth working-memory trajectory is compatible with the implemented schedule.

Algorithm 2 Importance-Driven Trickle-Down Consolidation

Require: EMA decay $\beta = 0.99$, transfer rates α_l, cascade ρ_l, snapshot refresh flag s.

Phase 1: Refresh empirical Snapshot Fisher only if s is true
if s **then**
  Temporarily enable gradients for all tiers to estimate importance.
  Zero gradients.
  Compute Snapshot Loss: $\mathcal{L}_{\text{snap}} \leftarrow \text{CrossEntropy}(z_{\text{new}}, y_{\text{active}})$
  Compute snapshot gradients of $\mathcal{L}_{\text{snap}}$
  $F_M \leftarrow \beta F_M + (1-\beta)\sum_{l=0}^{L-1}(\nabla_{M_{(l)}}\mathcal{L}_{\text{snap}})^2$
  $F_\Phi \leftarrow \beta F_\Phi + (1-\beta)\sum_{l=0}^{L-1}(\nabla_{\Phi_{(l)}}\mathcal{L}_{\text{snap}})^2$
  Restore LTM locks: refreeze all tiers where $l > 0$. Zero gradients.
end if
Phase 2: Adaptive thresholding & in-place Cartesian transfer (every step)
for $l = 0$ **to** $L-2$ **do**
  1. Calculate Joint Importance
  $P_M \leftarrow \text{Sigmoid}(\text{Standardize}(F_M))$
  $P_\Phi \leftarrow \text{Sigmoid}(\text{Standardize}(F_\Phi))$
  $P_{\text{joint}} \leftarrow \max(P_M, P_\Phi)$ {Critical if magnitude *or* phase is critical}
  2. Controller Update
  Empirical Pass Rate $E_l \leftarrow \text{Mean}(\mathbb{I}(P_{\text{joint}} > \gamma_l))$ {Query-branch scores, averaged over blocks}
  $\gamma_l \leftarrow \text{Clamp}(\gamma_l + 0.1 \times (E_l - \rho_l),\ 0.05, 0.99)$
  3. Cartesian Transformation & Safe Transfer
  $\mathcal{I}_{(l)} \leftarrow \text{ReLU}(P_{\text{joint}} - \gamma_l)$ {Soft-thresholding mask}
  $W_{(l),R} \leftarrow M_{(l)} \odot \cos(\Phi_{(l)}) \quad ; \quad W_{(l),I} \leftarrow M_{(l)} \odot \sin(\Phi_{(l)})$
  $\Delta W_R \leftarrow \alpha_l \cdot W_{(l),R} \odot \mathcal{I}_{(l)} \quad ; \quad \Delta W_I \leftarrow \alpha_l \cdot W_{(l),I} \odot \mathcal{I}_{(l)}$
  4. Apply sequential pair transfer; exact invariance only without clamp
  $d \leftarrow \text{sgn}(\lambda_{l+1}) \max(|\lambda_{l+1}|, 10^{-4})$ {Sign-preserving, with zero-sign convention}
  Ratio $r \leftarrow \lambda_l / d$ {Exact if clamp inactive}
  $W_{(l+1),R} \leftarrow W_{(l+1),R} + (\Delta W_R \cdot r)$
  $W_{(l+1),I} \leftarrow W_{(l+1),I} + (\Delta W_I \cdot r)$
  $W_{(l),R} \leftarrow W_{(l),R} - \Delta W_R$
  $W_{(l),I} \leftarrow W_{(l),I} - \Delta W_I$
  5. Reprojection to polar (in place; origin phase is not unique)
  $M_{(l)} \leftarrow \sqrt{W_{(l),R}^2 + W_{(l),I}^2} \quad ; \quad \Phi_{(l)} \leftarrow \text{arctan2}(W_{(l),I}, W_{(l),R})$
  $M_{(l+1)} \leftarrow \sqrt{W_{(l+1),R}^2 + W_{(l+1),I}^2} \quad ; \quad \Phi_{(l+1)} \leftarrow \text{arctan2}(W_{(l+1),I}, W_{(l+1),R})$
end for

3. Safe Cartesian transfer divisor. When $|\lambda_{l+1}| < 10^{-4}$ the sign-preserving denominator clamp avoids an arbitrarily large transfer ratio. It changes the exact pairwise conservation identity in the clamped branch; observed clamp frequency was 2/227,940 transfers. We do not describe clamped events as strictly output-invariant.

4. Strict CIL Target Masking: In strictly exemplar-free CIL, calculating the loss over all historical logits using one-hot targets inevitably pushes the model to suppress ancestral classes (since they are treated as negative examples for the new task). As shown in Algorithm 1, we structurally isolate the task optimization by dynamically re-indexing the ground-truth targets ($y - |\mathcal{Y}_{1:t-1}|$) and evaluating $\mathcal{L}_{CE}$ exclusively against the newly initialized columns of the linear head (z_{new}). This restricts the supervised loss to the current task but does not guarantee an absence of interference in the shared features.

B Extended implementation details and configurations

This section provides the comprehensive implementation details, hyperparameter configurations, and architectural specifications reported for reproducing the experimental results presented in the main paper. It bridges the formal algorithmic logic established in Appendix A with the specific engineering realities of our codebase.

B.1 Model architecture and polar padding dynamics

Backbone Configuration. Following the standard Class-Incremental Learning (CIL) scenario for parameter-efficient methods, we employ a pre-trained Vision Transformer (ViT-B/16) as our generalized feature extractor. Specifically, we utilize the `vit_base_patch16_224` configuration provided by the `timm` library. The entire backbone remains permanently frozen ($\nabla\phi = 0$) throughout the lifetime of the model.

Adapter Insertion and Structural Initialization. Our HiPo adapters are systematically injected exclusively into the Query and Value branches of the fused (`qkv`) linear projection layer within every Transformer block of the ViT backbone; the Key branch is left unadapted. Separate adapter stacks are used for Query and Value. For a memory stack of depth $L = 5$, the initial superposition scale weights (denoted as λ_l in the main text) are statically initialized to 1.0 for all tiers. During optimization, these act as dynamically learned scalars to balance geometric superposition.

Frequency Domain Padding. A critical, non-obvious requirement of our architecture involves matching the orthogonal sequence length to the block-diagonal partitions. The ViT-B/16 embedding dimension is $D = 768$. Applying the 1D Real Fast Fourier Transform (rFFT) across this dimension yields $F = \lfloor \frac{768}{2} \rfloor + 1 = 385$ complex frequency channels. Our fixed reference configuration uses a block size of $S = 32$. Because 385 is not neatly divisible by 32, we explicitly zero-pad the frequency dimension to the nearest multiple ($32 \times 13 = 416$) before applying the block-diagonal transformation. This yields exactly 13 discrete 32×32 complex blocks per layer. Following the Cartesian matrix multiplication, the feature tensor is truncated back to 385 channels before applying the inverse rFFT.

B.2 Hyperparameters and optimization schedule

To improve reproducibility, Table 4 comprehensively lists the exact hyperparameter suite utilized across all benchmarks.

As discussed in Appendix C, while standard spatial LoRA variants typically leverage adaptive optimizers (e.g., Adam), we use Stochastic Gradient Descent (SGD) with momentum to avoid the additional optimizer-state transformations that would be needed when using adaptive moment-based optimizers across in-place transfers. To facilitate rapid episodic assimilation within a concise 15-epoch training window, we employ an aggressive, near-continuous step decay schedule. The learning rate is multiplied by $\gamma = 0.9$ at almost every epoch milestone.

B.3 Detailed dataset statistics and preprocessing

To make baseline evaluations comparable, we use common class-incremental data splits and augmentation pipelines while retaining each method's published training schedule.

Dataset Statistics:

- **CIFAR-100:** 100 natural image classes, natively 32×32 resolution. 50,000 training images and 10,000 testing images.
- **CUB-200-2011 (Caltech-UCSD Birds):** 11,788 photographs from 200 bird species, with an official split of 5,994 training and 5,794 testing images. If a separate processed CIL split is used, its counts must be taken from the experimental data files.
- **ImageNet-R:** 30,000 renditions of 200 ImageNet classes, including art, cartoons, and sketches. The reported 24,000 training and 6,000 testing images refer to the derived CIL split, not an official training/test partition of ImageNet-R.
- **ImageNet-A:** 7,500 naturally occurring adversarial examples from 200 ImageNet classes. The original release is a test-only robustness benchmark with no official training split. For

Table 4: Comprehensive hyperparameter configuration for HiPo

Hyperparameter	**Value**
Optimization & Training (per episodic task)	
Optimizer	SGD (Momentum = 0.9)
Base Learning Rate (LR)	0.01
Weight Decay	0
Batch Size	16
Training Epochs	15
LR Decay Factor (γ)	0.9
LR Decay Milestones	$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]$
HiPo Architecture & Hierarchy	
Stack Depth (L)	5
Complex Block Size (S)	32
Initial Superposition Weights (λ)	$[1.0, 1.0, 1.0, 1.0, 1.0]$
Consolidation Mechanics	
Snapshot Fisher Frequency (f_{freq})	100 steps
Fisher EMA Decay Rate (β)	0.99
Controller Learning Rate (η)	0.1
Initial Threshold Cascade (γ_l at $\tau = 0$)	$[0.30, 0.70, 0.90, 0.95]$
Target Retention Cascade (ρ_l)	$[0.50, 0.25, 0.10, 0.05]$
Transfer Rate ($\alpha_l = 1/(f_0\gamma^l)$)	$f_0 = 100$, $\gamma = 10$

the class-incremental experiment, training and evaluation subsets are constructed under the study's data-partition protocol; the original 7,500 images must not be reported as 12,000 samples.

Data Augmentation and Preprocessing: Because our pre-trained ViT-B/16 backbone natively expects 224×224 inputs, all images are resized accordingly. During the active training phase (Algorithm 1), we apply standard random resized cropping and random horizontal flipping to prevent overfitting on the relatively small episodic subsets. During Nearest Mean of Exemplars (NME) prototype extraction and final testing, images are resized to 256×256 followed by a fixed 224×224 center crop. All inputs are normalized using standard ImageNet channel statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$).

B.4 Benchmark

For a sequence of $N = 10$ tasks on a 200-class dataset, the shuffled label space is rigidly partitioned into chunks of 20 disjoint classes per task (`init_cls=20`, `increment=20`). The model evaluates performance strictly on the aggregated test sets of all tasks seen up to the current sequence boundary, with zero access to episodic memory buffers (`memory_size=0`).

B.5 Nearest mean of exemplars (NME) protocol

NME is a standard classifier used here as a complementary probe of representation geometry. It does not have a guaranteed advantage over the learned classifier, and comparisons are head-matched whenever used for ranking methods. Specifically, at the conclusion of each task $\mathcal{T}_t$, the network is explicitly locked in evaluation mode, and the training dataset of the newly acquired classes is passed through the network to extract the spatial token representations. A class prototype is then defined as the mean vector of these extracted features, which is subsequently permanently frozen and appended to the historical prototype list. During inference, rather than computing distances using raw feature magnitudes, both the extracted query feature vector (f_q) and the stored class prototypes (p_c) are strictly L_2-normalized across the feature dimension. Classification is finally assigned based on the minimum squared Euclidean distance:

$$\arg\min_{c \in \mathcal{Y}_{1:t}} \left\| \frac{f_q}{\|f_q\|_2} - \frac{p_c}{\|p_c\|_2} \right\|_2^2 \tag{14}$$

Normalization makes the distance comparison depend on feature-vector direction, independently of the polar phases of adapter weights.

B.6 Hardware configuration and computational footprint

Our framework is explicitly designed to be hardware-efficient, bypassing the massive memory overheads associated with rehearsing historical data or computing full-network Fisher approximations.

All individual sequence runs (including baseline comparisons) were executed on a single NVIDIA P100 GPU (16GB VRAM) running PyTorch, with data loading parallelized across 4 CPU workers. Because the pre-trained ViT backbone is permanently frozen, and the episodic training utilizes a batch size of 16, peak GPU memory utilization comfortably remains under 10 GB. The temporary memory expansion occurs during the Snapshot Fisher refresh (every 100 steps), where auto-differentiation briefly tracks gradients for the deeper L tiers. HiPo uses frozen-backbone adaptation and a 15-epoch training schedule; baseline methods retain their published schedules. A complete 10-task HiPo sequence on ImageNet-R (from Task 1 through Task 10) completes in approximately 1.5 to 2 hours on a single P100, within this reported experimental setup.

C Numerical stability and optimization manifold

This section describes the Cartesian forward mapping, safe-divisor clamp, and optimizer choice used to implement polar parameters and hierarchical transfers.

C.1 Resolving the polar origin singularity

A fundamental challenge of explicitly optimizing neural network weights in polar coordinates is the coordinate singularity at the origin. The angle is non-unique at zero magnitude. Differentiating an inverse-angle map at the origin may be problematic, but routine task optimization uses the Cartesian forward map instead.

During task optimization, the forward computation maps polar parameters into Cartesian weights without differentiating through an inverse-angle function:

$$W_R = M \odot \cos(\Phi), \quad W_I = M \odot \sin(\Phi) \tag{15}$$

By applying the chain rule to this continuous Cartesian mapping, the gradient of the loss $\mathcal{L}$ with respect to the phase explicitly evaluates to:

$$\frac{\partial \mathcal{L}}{\partial \Phi} = \frac{\partial \mathcal{L}}{\partial W_R}(-M \odot \sin(\Phi)) + \frac{\partial \mathcal{L}}{\partial W_I}(M \odot \cos(\Phi)) \tag{16}$$

For bounded Cartesian gradients, the norm of the phase gradient is bounded by a constant times $|M|$; this is magnitude-proportional damping, not evidence that the working-memory magnitudes approach zero or all phase updates stop. We do not differentiate through inverse-angle reconstruction during ordinary task training.

C.2 Geometric superposition and safe ratio scaling

During the trickle-down consolidation (Algorithm 2), knowledge is transferred from a shallow memory tier (l) to a deeper tier ($l+1$). For the unclamped transfer to conserve the effective adapter weight, the Cartesian mass ΔW must be scaled relative to the learned superposition weights (λ).

Equating the output geometry yields the required balancing ratio:

$$\lambda_l W_{(l)} + \lambda_{l+1} W_{(l+1)} = \lambda_l (W_{(l)} - \Delta W) + \lambda_{l+1} \left(W_{(l+1)} + \Delta W \cdot r \right) \implies r = \frac{\lambda_l}{\lambda_{l+1}} \tag{17}$$

However, because λ_{l+1} is a dynamically optimized scalar, it is numerically possible for it to fluctuate near zero, which would cause an arithmetic overflow when computing r. For numerical stability, the

implementation applies a sign-preserving clamp before division, which compromises exact invariance only when active. Specifically, the transfer ratio is computed as:

$$\text{Safe_Dest} = \begin{cases} \lambda_{l+1} & \text{if } |\lambda_{l+1}| > 10^{-4} \\ 10^{-4} \times \text{sign}(\lambda_{l+1}) & \text{otherwise} \end{cases} \tag{18}$$

$$r = \frac{\lambda_l}{\text{Safe_Dest}} \tag{19}$$

Let $d_l = \text{Safe_Dest}$. After a clamped transfer the effective pairwise change is $\lambda_l \Delta W_{(l)}(\lambda_{l+1}/d_l - 1)$. When clamped, $\lambda_{l+1}/d_l \in [0, 1]$, so a portion of the effective transferred contribution is discarded rather than amplified. In the unclamped branch the change vanishes exactly. The clamp was observed on 2 of 227,940 transfers.

C.3 Magnitude-proportional phase damping

After Cartesian mass transfer, the updated real and imaginary weights are converted to magnitude and phase coordinates. At zero magnitude the phase is not unique, so practical reprojection uses a convention for this case; the phase-sensitivity bound below applies at any fixed magnitude.

The per-coordinate phase-change bound $|M(e^{i(\Phi+\Delta\Phi)} - e^{i\Phi})| \leq 2|M|$ and the phase-gradient bound characterize sensitivity at a fixed magnitude. In the measured dynamics, magnitude declines only gradually as transfer and optimization balance, and the phase gradients remain small (within 0–0.0175 over 25,000 steps). These observations support damping rather than complete paralysis or a general guarantee against forgetting.

C.4 Optimization manifold compatibility (Adam vs. SGD)

As documented in Appendix B, the HiPo framework utilizes Stochastic Gradient Descent (SGD) with momentum, deliberately eschewing adaptive optimizers like Adam. SGD was used throughout the reported experiments because its optimizer state is simpler to manage across direct parameter transfers.

Adam independently maintains historical first (m) and second (ν) moment estimates for every parameter. During consolidation, we move parameter contributions between tiers and re-express the updated weights in polar coordinates. If Adam were utilized, the historical moment accumulators would remain aggressively conditioned on the pre-transfer parameter states. Without a suitable state-transformation rule, Adam's moments could become inconsistent with modified parameters. We do not claim that Adam necessarily fails without an optimizer-controlled comparison.

Conversely, SGD with momentum updates weights utilizing a much shorter memory horizon and lacks parameter-specific variance denominators. We report this as an implementation choice rather than a theorem that SGD is uniquely compatible with the proposed transfer.

D Theoretical derivations & proofs

D.1 Preliminaries, notation, and assumptions

Let $f(\cdot; \Theta)$ denote the neural network mapping. Following the notation of Section 3.2, the incremental parameters for a localized frequency block k across the hierarchical stack of depth L are collapsed into an effective affine transformation during inference:

$$W_{\text{eff}}^{(k)} = \sum_{l=0}^{L-1} \lambda_l W_{(l)}^{(k)} = \sum_{l=0}^{L-1} \lambda_l \left(M_{(l)}^{(k)} \odot \cos(\Phi_{(l)}^{(k)}) + i \left[M_{(l)}^{(k)} \odot \sin(\Phi_{(l)}^{(k)}) \right] \right) \tag{20}$$

where $M_{(l)}^{(k)}, \Phi_{(l)}^{(k)} \in \mathbb{R}^{S \times S}$ are the polar magnitude and phase parameters for tier l in block k, and $\lambda_l \in \mathbb{R}$ are the continuous scalar layer superposition weights.

Assumption 1 (Conditions used in the analyses)**.** *For the stated local bounds, suppose (i) the input and local Cartesian gradients are bounded by finite constants, and (ii) a single-task adapter*

displacement has bounded norm. For the conditional *anchoring result below, additionally suppose that effective deep-tier coefficients are fixed, all relevant update contributions are retained in the anchor, the aggregate expected cross-term energy is nonnegative, and the old-feature covariance is nondegenerate over the anchor subspace. These conditions are not enforced by the numerical clamp or by the general training algorithm. Empirically, per-task update magnitudes decline, while the deep-tier coefficients shrink; see Appendix F.*

D.2 Magnitude-proportional phase sensitivity

We characterize the sensitivity of a complex weight to phase changes at a fixed magnitude. This pointwise statement does not imply that consolidation makes magnitudes vanish in practice, or that past-task predictions are preserved.

Lemma 1 (Geometric Bounds of Polar Reprojection)**.** *Under the continuous Cartesian forward mapping $W = M \odot \cos(\Phi) + i(M \odot \sin(\Phi))$, under bounded Cartesian gradients the phase gradient scales at most linearly with $|M|$. For a fixed magnitude and any phase perturbation $\Delta\Phi$, the induced weight change is bounded by $2|M|$. This bounds phase sensitivity, not forgetting.*

Proof. Let $W_{R,j} = M_j \cos(\Phi_j)$ and $W_{I,j} = M_j \sin(\Phi_j)$ represent the Cartesian components of a single scalar weight $j \in \{1, \ldots, S^2\}$. The gradient of the loss with respect to the phase can be expressed as the dot product of the Cartesian gradient vector and the trigonometric derivatives. Applying the multivariate chain rule:

$$\frac{\partial \mathcal{L}}{\partial \Phi_j} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial W_{R,j}} \\ \frac{\partial \mathcal{L}}{\partial W_{I,j}} \end{bmatrix} \cdot \begin{bmatrix} -M_j \sin(\Phi_j) \\ M_j \cos(\Phi_j) \end{bmatrix} \tag{21}$$

If each Cartesian gradient component is bounded by K (an explicit local condition of Assumption 1), we apply the Cauchy-Schwarz inequality:

$$\begin{aligned} \left| \frac{\partial \mathcal{L}}{\partial \Phi_j} \right| &\leq \sqrt{\left(\frac{\partial \mathcal{L}}{\partial W_{R,j}} \right)^2 + \left(\frac{\partial \mathcal{L}}{\partial W_{I,j}} \right)^2} \sqrt{M_j^2 \sin^2(\Phi_j) + M_j^2 \cos^2(\Phi_j)} \\ &\leq \sqrt{2K^2} \cdot M_j = M_j \sqrt{2} K \end{aligned} \tag{22}$$

Therefore, as $M_j \to 0$, the exact analytical gradient strictly vanishes ($\nabla_{\Phi_j} \mathcal{L} \to 0$).

To bound sensitivity at fixed magnitude, consider the parametric change ΔW_j in the complex Cartesian domain caused by an arbitrary phase shift $\Delta\Phi_j$:

$$\begin{aligned} |\Delta W_j|^2 &= \left| \left[M_j e^{i(\Phi_j + \Delta\Phi_j)} \right] - \left[M_j e^{i\Phi_j} \right] \right|^2 \\ &= M_j^2 \left| \cos(\Phi_j + \Delta\Phi_j) - \cos(\Phi_j) + i(\sin(\Phi_j + \Delta\Phi_j) - \sin(\Phi_j)) \right|^2 \\ &= M_j^2 \left[(\cos(\Phi_j + \Delta\Phi_j) - \cos(\Phi_j))^2 + (\sin(\Phi_j + \Delta\Phi_j) - \sin(\Phi_j))^2 \right] \\ &= M_j^2 \left[2 - 2\cos(\Delta\Phi_j) \right] \leq 4M_j^2 \end{aligned} \tag{23}$$

Taking the square root, the maximum parametric distortion caused by any phase jump is strictly bounded by $2M_j$. The bound shrinks for a small M_j; it says nothing about whether consolidation makes this magnitude small during training. □

Remark 1 (Cartesian Geometric Invariance)**.** *For an* unclamped *pairwise transfer, subtracting ΔW from tier l and adding $(\lambda_l / \lambda_{l+1})\Delta W$ to tier $l+1$ leaves $\sum_j \lambda_j W_{(j)}$ unchanged. A sequential composition of such pairwise operations is also invariant. In the clamped branch, a residual change remains as shown in Appendix C.2; the property is conditional, not absolute.*

D.3 Relative structural anchoring (conditional)

Proposition 1 (Conditional relative structural shift)**.** *Let $u_i^{(k)}$ be the effective contributions transferred into a fixed-coefficient long-term anchor $C_t^{(k)} = \sum_{i=1}^{t} u_i^{(k)}$. Suppose that $\mathbb{E}\|u_i^{(k)}\|_F^2 \geq c > 0$, their* aggregate *expected pairwise inner-product sum is nonnegative, and the expected old-feature energy is at least $a\mathbb{E}\|C_t^{(k)}\|_F^2$ for $a > 0$. If a subsequent task produces an update whose expected*

feature-shift energy is bounded by a constant b, then the ratio of these expected energies is at most $b/(act) = \mathcal{O}(1/t)$, and its square root is $\mathcal{O}(1/\sqrt{t})$. This is a conditional representation-energy ratio, not a classification-forgetting bound.

Proof. By expansion,

$$\mathbb{E}\|C_t^{(k)}\|_F^2 = \sum_{i=1}^{t} \mathbb{E}\|u_i^{(k)}\|_F^2 + 2\sum_{i<j} \mathbb{E}\langle u_i^{(k)}, u_j^{(k)}\rangle_F \geq ct. \tag{24}$$

The assumed nondegeneracy gives $\mathbb{E}\|\tilde{X}^{(k)}C_t^{(k)}\|_2^2 \geq act$, while the new-task feature displacement has energy at most b. Dividing yields the stated upper bound on their ratio; taking the square root gives the relative root-mean-square scaling. The argument does not require each pairwise cross term to be nonnegative, only their aggregate sum. □

Remark 2 (Empirical observations)**.** *Proposition 1 characterizes structural anchoring under fixed coefficients and accumulated anchor energy. In practice, HiPo learns adaptive deep-tier λ values that decrease during training, resulting in a different consolidation regime. Empirically, stability is characterized by declining per-task update magnitudes and consistently low feature drift, as detailed in Appendix F.*

D.4 Spatial dispersion and interference bounds in the Fourier domain

Proposition 2 (Coordinate-wise dispersion of a one-sided Fourier update)**.** *Let $v \in \mathbb{C}^F$ be the one-sided spectrum of a real D-dimensional feature displacement, with at most S nonzero bins. Under orthonormal inverse real FFT reconstruction,*

$$\left\| \mathrm{irFFT}(v; n = D, \mathrm{norm} = \mathrm{ortho}) \right\|_\infty \leq 2\sqrt{\frac{S}{D}}\, \|v\|_2. \tag{25}$$

This bounds an individual spatial coordinate for a localized update and does not bound the total interference accumulated across tasks.

Proof. The real inverse transform reconstructs non-DC/non-Nyquist bins with conjugate pairs. For every spatial coordinate, the inverse-transform sum has magnitude at most $2\|v\|_1/\sqrt{D}$, where the coefficient 2 is conservative for the real DC and Nyquist terms. Since v is at most S-sparse, Cauchy–Schwarz yields $\|v\|_1 \leq \sqrt{S}\|v\|_2$, proving the claim. In the one-sided representation Parseval's identity is weighted: interior bins count twice in the full Hermitian spectrum, so $\|\mathrm{irFFT}(v)\|_2 \leq \sqrt{2}\|v\|_2$ rather than equality with the unweighted one-sided norm. Removing zero-padding is a non-expansive coordinate projection. Neither property changes the $\sqrt{S/D}$ dependence of the coordinate bound. □

Remark 3 (Scope of dispersion)**.** *The proposition characterizes how orthogonal mixing distributes an S-sparse block-local update across feature dimensions while preserving its total energy. For updates spanning multiple blocks or accumulating over successive tasks, the resulting interference also depends on their combined structure. The ablation results and measured feature drift provide complementary empirical evidence of how spatial dispersion contributes to reducing forgetting.*

D.5 Theoretical Insight: Fisher-Guided structural isolation

According to Elastic Weight Consolidation (EWC) theory, the expected loss degradation on ancestral tasks caused by overwriting an established parameter w is governed by the second-order Taylor expansion: $\Delta\mathcal{L} \approx \frac{1}{2}F_w(\Delta w)^2$, where F_w is the diagonal of the Fisher Information Matrix approximating the local Hessian.

While EWC uses Fisher information to regularize parameter updates, HiPo uses normalized empirical Snapshot Fisher scores to gate transfer between tiers. Higher scores prioritize parameters for consolidation rather than directly penalizing their gradients. This provides an importance-sensitive routing heuristic; the local Fisher approximation alone does not establish a guarantee against later interference.

D.6 Convergence and stability of the adaptive consolidation controller

A proportional feedback controller adapts the retention threshold γ_l to the observed importance-score distribution. Its theoretical stability result assumes regularity and a decaying step size, while the fixed-step implementation tracks the target approximately.

Lemma 2 (Adaptive Threshold Stability via Stochastic Approximation)**.** *For a continuous stationary score distribution with positive density near an interior target quantile and a standard decaying stochastic-approximation step size, the feedback controller targets the quantile with upper-tail probability ρ_l. With fixed η, exact expectation convergence is not guaranteed.*

Proof. Let P_{joint} be the standardized, Sigmoid-projected Fisher criticality score defined in Section 3.4. Its probability density function (PDF), denoted $p(\gamma_l)$, is continuous. The corresponding cumulative distribution function (CDF) is $F(\gamma_l) = \mathbb{P}(P_{\text{joint}} \leq \gamma_l)$.

During training, $E_l = \mathbb{E}_{\text{batch}}[\mathbb{I}(P_{\text{joint}} > \gamma_l)]$ is an empirical expectation drawn from a snapshot batch, acting as an unbiased estimator for $1 - F(\gamma_l)$. Substituting the true expectation into the controller update rule:

$$\gamma_l \leftarrow \gamma_l + \eta(1 - F(\gamma_l) - \rho_l) \tag{26}$$

Consider the objective function designed to minimize the discrepancy between the CDF and the target distribution rate:

$$\mathcal{L}(\gamma_l) = \frac{1}{2}\Big(F(\gamma_l) - (1 - \rho_l)\Big)^2 \tag{27}$$

The gradient of this objective with respect to the threshold is:

$$\nabla_{\gamma_l}\mathcal{L} = \Big(F(\gamma_l) - (1 - \rho_l)\Big)p(\gamma_l) \tag{28}$$

By factoring out the strictly positive PDF, the controller update can be rewritten exactly as:

$$\gamma_l \leftarrow \gamma_l - \frac{\eta}{p(\gamma_l)}\nabla_{\gamma_l}\mathcal{L} \tag{29}$$

This demonstrates that our heuristic feedback controller is mathematically equivalent to Gradient Descent preconditioned by a strictly positive scalar ($1/p(\gamma_l)$). The projection constrains the threshold, but by itself does not imply a positive lower bound on the score density. Such a bound must be separately assumed on the relevant quantile neighborhood.

Under a stationary Fisher-score distribution with a unique target quantile and positive density in its neighborhood, the controller admits a standard stochastic-approximation analysis when the step sizes satisfy the Robbins–Monro conditions ($\sum_n \eta_n = \infty$, $\sum_n \eta_n^2 < \infty$). In practice, HiPo employs a fixed step size to adapt the retention threshold as the Fisher-score distribution evolves during training. This enables responsive importance-based consolidation, while the theoretical convergence analysis characterizes the stationary, diminishing-step setting. □

E Detailed ablation study

Section 4.4 reports the primary NME ablations. Table 5 also reports results with the parametric classifier (CNN) for each task sequence. We then describe the architectural changes in each ablation and the conclusions they support.

E.1 Isolating core framework mechanisms

Ablation 1: Spatial-Cartesian (rank-matched LoRA). This adapter preserves the fixed memory stack, Fisher-gated transfer, learned tier weights and SGD schedule, but omits the Fourier mixing and polar coordinates. The frequency-polar pair raises ImageNet-A NME from 63.15 to 64.39 (+1.24 points), while spatial-Cartesian remains competitive on ImageNet-R. Thus, the hierarchy alone accounts for useful performance, and the geometric refinement has a limited additional effect in-distribution.

Ablation 2: Spatial-Polar. Removing Fourier mixing while retaining a polar-style real spatial parameterization yields 49.78 NME on ImageNet-A versus 63.15 for spatial-Cartesian. The strong

Table 5: Comprehensive ablation results detailing both Parametric Head (CNN) and Exemplar Prototype (NME) evaluations across all benchmark sequences. Results are reported as Mean ± Standard Deviation over 5 independent runs. The full-model entries match Table 1. Best results are in bold, and second-best results are underlined.

Ablation Variant	Eval	IN-R (5 tasks)		IN-R (10 tasks)		IN-R (20 tasks)		IN-A (10 tasks)	
		AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc
HiPo (Full)	*CNN*	83.87 ±0.25	**80.45 ±0.37**	83.56 ±0.41	78.91 ±0.14	82.37 ±0.44	77.46 ±0.36	65.58 ±0.13	56.41 ±0.36
	NME	**83.90 ±0.25**	80.38 ±0.19	**83.75 ±0.40**	79.21 ±0.45	**83.37 ±0.32**	79.00 ±0.37	71.37 ±0.88	**64.39 ±0.52**
Spatial-Cartesian (LoRA)	*CNN*	83.03 ±0.22	79.94 ±0.50	83.48 ±0.58	78.78 ±0.55	82.25 ±0.49	77.38 ±0.70	63.67 ±0.27	54.23 ±0.41
	NME	83.43 ±0.39	79.88 ±0.52	83.35 ±0.81	79.11 ±0.85	83.28 ±0.13	78.34 ±0.55	70.97 ±0.67	63.15 ±0.56
w/o Orthogonal Proj. (Spatial-Polar)	*CNN*	77.56 ±0.42	71.78 ±0.63	78.33 ±0.88	72.35 ±0.64	77.34 ±0.45	70.26 ±0.28	58.34 ±0.68	49.45 ±0.64
	NME	78.05 ±0.43	71.58 ±0.43	79.19 ±0.89	72.93 ±0.60	78.64 ±0.66	71.45 ±0.31	59.45 ±0.58	49.78 ±0.55
w/o Polar Disentanglement (Freq.-Cart.)	*CNN*	83.46 ±0.58	79.89 ±0.10	83.29 ±0.62	78.66 ±0.23	82.16 ±0.68	77.06 ±0.26	65.27 ±1.29	56.55 ±0.50
	NME	83.02 ±0.37	79.89 ±0.27	83.04 ±0.59	**79.51 ±0.55**	83.13 ±0.61	78.91 ±0.89	71.38 ±0.52	63.87 ±0.58
w/o Hierarchical Consolidation	*CNN*	83.27 ±0.58	80.16 ±0.25	83.16 ±0.48	78.44 ±0.88	82.07 ±0.53	77.14 ±0.31	64.99 ±1.00	55.23 ±0.35
	NME	83.34 ±0.42	80.36 ±0.31	82.78 ±0.38	78.66 ±0.29	83.21 ±0.49	**79.11 ±0.33**	71.01 ±1.16	63.35 ±0.28
w/o Cartesian-Invariant Transfer	*CNN*	82.53 ±0.29	79.04 ±0.81	82.98 ±0.34	78.82 ±0.63	81.76 ±0.71	77.08 ±0.51	65.52 ±1.15	56.35 ±1.31
	NME	82.71 ±0.16	79.34 ±0.77	82.52 ±0.24	78.90 ±0.12	82.87 ±0.50	78.95 ±0.53	**71.59 ±0.74**	62.87 ±0.75

negative interaction indicates that this polar parameterization is poorly matched to raw unordered channels, not that all spatial adapters require an rFFT.

Ablation 3: Frequency-Cartesian. Retaining Fourier mixing but removing polar coordinates yields 63.87 NME / 56.55 CNN on ImageNet-A, versus 64.39 / 56.41 for HiPo. The polar effect is small and changes sign across evaluation heads. On ImageNet-R 10T, frequency-Cartesian achieves higher NME final accuracy than HiPo (79.51 versus 79.21); this difference is not statistically significant under the paired comparison.

Ablation 4: No hierarchical consolidation. The no-transfer variant obtains 63.35 ImageNet-A NME versus 64.39 for full HiPo, but is close to or slightly above it on some ImageNet-R settings. This suggests a regime-dependent benefit, not an essential role in every task sequence. The fixed-depth hierarchy's separate capacity accounting is discussed in the main results.

Ablation 5: Naive polar arithmetic. Adding angles directly is not equivalent to adding their represented complex weights. This ablation reaches 62.87 NME on ImageNet-A versus 64.39 for the Cartesian-transfer variant, supporting the algebraic choice of complex addition for conservation when unclamped.

Statistical interpretation. Paired seed-level t-tests (five seeds, four degrees of freedom) compare HiPo with four component-removal ablations. The orthogonal-projection effect is significant in all four benchmark settings; polar removal is significant only on ImageNet-R 5T, hierarchical removal only on ImageNet-A, and transfer removal on ImageNet-R 5T and ImageNet-A. Because these comprise 16 comparisons, nominal significance must be treated cautiously. Projection and the significant ImageNet-A effects persist after multiplicity adjustment; exact adjusted p values are not tabulated.

E.2 Sensitivity analysis of the hierarchical stack and consolidation cascade

We examine sensitivity to two structural hyperparameters: the memory-stack depth L and the initial threshold cascade γ_l. Table 6 reports performance across all benchmark sequences for both prediction heads.

E.2.1 Retention schedule

Configurations. The cascades varied here are the initial thresholds γ_l (the `lora_thresholds` setting), while the target retention rates ρ_l of the controller are fixed at $[0.50, 0.25, 0.10, 0.05]$. The reference cascade is $[0.30, 0.70, 0.90, 0.95]$, compared with more restrictive $[0.50, 0.80, 0.95, 0.99]$ and more permissive $[0.10, 0.30, 0.50, 0.70]$ variants. On ImageNet-A (NME), the two extremes give 62.79 and 62.91, versus 64.39 with the reference cascade. These results support sensitivity to transfer rate, but cannot by themselves isolate the cause as noise absorption or depleted plasticity.

Table 6: Comprehensive sensitivity analysis of HiPo's hierarchical hyperparameters. We evaluate the impact of the initial threshold cascade (γ_l) and the total stack depth (L). Results are reported as Mean $\pm$ Standard Deviation over 5 independent runs, with the common reference configuration indicated at $L = 5$. Best results are in bold, and second-best results are underlined.

Ablation Variant	Eval	IN-R (5 tasks)		IN-R (10 tasks)		IN-R (20 tasks)		IN-A (10 tasks)	
		AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc
Reference Configuration ($L = 5$)	*CNN*	83.87 ±0.25	80.45 ±0.37	83.56 ±0.41	78.91 ±0.14	82.37 ±0.44	77.46 ±0.36	65.58 ±0.13	56.41 ±0.36
	NME	**83.90 ±0.25**	80.38 ±0.19	83.75 ±0.40	79.21 ±0.45	83.37 ±0.32	**79.00 ±0.37**	**71.37 ±0.88**	**64.39 ±0.52**
Consolidation Cascade Dynamics (initial γ_l)									
Strict Cascade $[0.5, 0.8, 0.95, 0.99]$	*CNN*	83.29 ±0.56	79.74 ±0.16	83.47 ±0.57	78.75 ±0.16	82.74 ±0.68	77.34 ±0.26	65.58 ±0.71	56.31 ±0.57
	NME	83.77 ±0.37	80.05 ±0.42	**83.83 ±0.51**	79.10 ±0.35	**83.59 ±0.59**	78.59 ±0.18	71.11 ±0.98	62.79 ±0.45
Permissive Cascade $[0.1, 0.3, 0.5, 0.7]$	*CNN*	83.26 ±0.68	79.96 ±0.01	83.68 ±0.52	78.96 ±0.43	82.54 ±0.69	76.78 ±0.20	65.62 ±0.80	56.39 ±0.38
	NME	83.69 ±0.41	79.98 ±0.38	83.81 ±0.49	**79.29 ±0.68**	83.36 ±0.53	78.56 ±0.29	71.14 ±1.20	62.91 ±0.38
Hierarchical Memory Stack Depth (L)									
Stack Depth: 1 Layer	*CNN*	83.08 ±0.31	80.27 ±0.14	83.04 ±0.38	78.15 ±0.30	81.86 ±0.61	76.31 ±0.33	64.63 ±0.90	55.31 ±0.42
	NME	83.22 ±0.26	80.39 ±0.21	82.88 ±0.49	78.36 ±0.15	82.37 ±0.52	77.96 ±0.02	71.02 ±1.11	62.99 ±0.51
Stack Depth: 3 Layer	*CNN*	83.63 ±0.34	80.28 ±0.47	83.26 ±0.47	78.53 ±0.61	82.12 ±0.64	77.44 ±0.27	65.13 ±0.44	56.23 ±0.11
	NME	83.29 ±0.37	80.63 ±0.48	83.10 ±0.50	78.57 ±0.14	82.76 ±0.54	78.94 ±0.21	71.21 ±0.89	64.07 ±0.44
Stack Depth: 7 Layer	*CNN*	83.20 ±0.24	**80.89 ±0.36**	83.79 ±0.59	78.85 ±0.07	82.28 ±0.78	77.40 ±0.22	65.01 ±1.29	56.39 ±0.26
	NME	82.47 ±0.32	79.66 ±0.41	83.67 ±0.47	79.25 ±0.29	83.42 ±0.50	78.74 ±0.26	70.79 ±0.96	62.95 ±0.21

E.2.2 Stack depth

Configurations. For $L \in \{1, 3, 5, 7\}$, ImageNet-A NME final accuracy is 62.99, 64.07, 64.39 and 62.95. Depth is not monotonically beneficial, and the 3- and 5-tier scores are not clearly separated by their reported standard deviations. Learned deep-tier coefficients decreased during training; thus an under-realized effective depth and a capacity–regularization tradeoff are plausible explanations, not established causal findings.

F Feature-space diagnostics and implementation checks

F.1 Direct feature-drift measurements

Table 7: Old-feature drift across task sequences (lower indicates less change).

Metric	IN-A 10T	IN-R 10T	IN-R 20T
Mean drift, all task pairs	0.256	0.376	0.370
Mean drift at final task	0.232	0.370	0.361
First-step drift, early tasks	0.222	0.284	0.264
First-step drift, late tasks	0.145	0.247	0.194

The results show that HiPo maintains stable feature representations as the task sequence grows. On ImageNet-R, the mean drift at the final task remains comparable between 10 and 20 tasks (0.370 vs. 0.361). Moreover, first-step drift decreases for later tasks across all settings, indicating progressively smaller changes to previously learned representations. A separate temporal analysis on ImageNet-R 20T reports a drift increase of 0.308 over the first ten tasks, followed by only 0.007 over the remaining nine transitions, suggesting a gradual stabilization of feature representations. Together, these observations provide empirical evidence of representation stability across longer task sequences.

F.2 Assumption and implementation ledger

- **Bounded and decaying updates.** Measured per-task update magnitudes decrease over the task sequence. This empirical trend does not itself prove a forgetting bound.
- **Cross-term condition.** Aggregate update cross-term energy satisfied the condition invoked by the conditional analysis; the stronger claim that every pairwise term is nonnegative was contradicted by approximately 40–45% negative pairs.
- **Anchor growth.** The fixed-λ premise of Proposition 1 does not hold during training: deep-tier coefficients decline and the purported anchor does not show growing energy.

- **Cartesian transfer clamp.** Only 2 out of 227,940 transfers used the safe-divisor clamp. The unclamped identity is exact algebraically, whereas the clamped branch can discard effective mass.
- **Fourier reconstruction.** For ViT-B/16, $D = 768$, one-sided length $F = 385$, padded length 416, and block size $S = 32$. We discard padded bins, then apply inverse real FFT with explicit length 768 and orthonormal scaling. Imaginary parts of DC and Nyquist bins do not affect the real inverse and receive no useful gradient.
- **Free configuration variables.** The full results use the same base hyperparameters across tasks and datasets. Only hierarchy depth L and the initial threshold cascade γ_l are emphasized as free structural hyperparameters; S sets block capacity, f_{freq} is a fixed Snapshot Fisher interval, and SGD is the reported optimizer.

F.3 RanPAC classifier-head controls

Table 8: Effect of RanPAC prediction-head variants. Each cell reports mean AAA / final Acc. RP denotes random projection; NCM denotes nearest-class-mean classification.

Benchmark	Full RanPAC	Without RP	NCM only
IN-R (5T)	82.95 / 79.65	80.17 / 76.23	78.76 / 74.77
IN-R (10T)	82.61 / 78.14	77.65 / 73.59	76.24 / 72.06
IN-R (20T)	79.77 / 75.51	74.66 / 70.79	70.48 / 65.79
IN-A (10T)	70.17 / 58.33	69.07 / 57.23	60.26 / 49.87
CIFAR-100 (10T)	93.11 / 89.68	91.65 / 87.09	85.14 / 80.26
CUB-200 (10T)	86.64 / 82.08	83.36 / 78.99	80.78 / 77.49

Table 8 illustrates the prediction head's substantial contribution, particularly on in-distribution datasets where full RanPAC outperforms its NCM-only counterpart. These comparisons isolate the effect of the head rather than the adapter architecture.

G Limitations and broader impacts

While HiPo provides a geometric dispersion bound and experimental evidence under the evaluated CIL protocol, it introduces specific algorithmic trade-offs and real-world implications.

G.1 Complexity analysis and algorithmic limitations

Inference and Training Complexity: During inference, the L-tier stack algebraically collapses into a single effective block-diagonal matrix. Thus, inference time is independent of depth L, scaling competitively at $\mathcal{O}(B \cdot T \cdot D(\log D + S))$, where D is the embedding dimension, T the sequence length, and S the block size. During ordinary task optimization, gradients are tracked primarily for the working-memory tier ($l = 0$); Snapshot Fisher refreshes also require temporary gradient computations for deeper tiers.

The Snapshot Memory Spike: A key limitation arises during the Empirical Snapshot Fisher event (every f_{freq} steps). Computing the Fisher Information Matrix requires temporarily unfreezing the entire Long-Term Memory (LTM) stack. This induces a localized compute spike of $\mathcal{O}(B \cdot T \cdot F \cdot S \cdot L)$ and a peak GPU VRAM spike of $\mathcal{O}(B \cdot T \cdot F \cdot L)$ to cache intermediate activations. While Snapshot Fisher overhead is amortized over f_{freq} steps, the absolute $\mathcal{O}(L)$ memory spike acts as a hard hardware constraint for edge devices.

Optimizer Constraints (Adam Incompatibility): The reported implementation uses SGD with momentum. Adaptive optimizers (e.g., Adam) independently track historical second moments (variance) for every parameter. During discrete consolidation, parameter mass is moved between tiers and reprojected into polar coordinates. An adaptive optimizer would require consistent handling of its state after in-place weight transfers; such alternatives have not been benchmarked here.

The Unlearning Bottleneck: The conditional anchor proposition does not establish a forgetting bound, and the measured deep-tier contributions do not exhibit the required anchor growth. Because

HiPo is designed to retain selected updates, unlearning erroneous or corrupted information may be challenging. Targeted unlearning has not been evaluated.

G.2 Broader impacts

Privacy and Environmental Efficiency: The CUB-200-2011 dataset maintainers caution that some bird images overlap with ImageNet. Accordingly, results using an ImageNet-pretrained backbone should not be interpreted as measuring pretraining-independent generalization without a separate overlap audit (40).

A primary societal benefit of HiPo is its strict exemplar-free (zero-buffer) nature. Exemplar-free learning avoids retaining old task images in an episodic replay buffer, which can reduce one source of stored information. This alone does not guarantee compliance with privacy regulations, and environmental savings were not measured.

Robustness in Safety-Critical Deployments: ImageNet-A experiments suggest potential robustness benefits under a particular distribution shift. No safety guarantee, certified robustness, or evidence of readiness for medical or autonomous deployment is established.

Risks of Bias Entrenchment and Dual-Use: Long-term retention could also preserve undesirable correlations or biases learned from earlier tasks. Whether hierarchical consolidation makes later bias correction or targeted unlearning harder remains untested; deployment in sensitive settings would require suitable bias evaluation and correction procedures.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the core contributions (frequency-domain polar adaptation, hierarchical stacked memory, and importance-driven consolidation), which are directly supported by the formal methodology (Section 3) and experimental results (Section 4).

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss algorithmic limitations, including the temporary "Snapshot Memory Spike", optimizer constraints (Adam incompatibility), and the "Unlearning Bottleneck" in Appendix E.1.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The formal assumptions are clearly stated in Appendix D.1, and complete proofs for all theoretical claims (Lemmas 1 & 2, Propositions 1 & 2) are provided in Appendices D.2 through D.6.

4. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The datasets used are strictly standard public benchmarks. While we thoroughly describe our algorithms and hyperparameters to ensure reproducibility without code, the raw source code is not included in this anonymized submission but will be released upon publication.

5. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix B explicitly details the model architecture, hyperparameter configuration (Table 3), optimization schedule (SGD with momentum), and the deterministic class ordering protocols used for all CIL benchmarks.

6. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main results and ablation studies (Tables 1 and 2) report the mean and standard deviation across 5 independent runs to capture the variance induced by random initialization and episodic training.

7. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix B.6 details the exact hardware configuration (a single NVIDIA P100 GPU with 16GB VRAM, 4 CPU workers) and reports the estimated execution time and peak memory footprint required for the experiments.

8. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics, conducting fundamental research on continual learning without violating privacy or engaging in inherently harmful applications.

9. **Broader impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [Yes]

   Justification: We discuss broader impacts in Appendix E.2, addressing positive impacts like privacy preservation (zero-buffer CIL) and computational efficiency, as well as potential negative risks such as bias entrenchment and dual-use surveillance.

10. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes a generic continual learning optimization framework and does not release any high-risk foundational models or scraped datasets.

11. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The pre-trained ViT backbone and standard academic datasets (CIFAR-100, CUB-200, ImageNet-R, ImageNet-A) are properly cited in the main text and bibliography in accordance with academic standards.

12. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces a novel algorithmic framework but does not introduce or release any new proprietary datasets.

13. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research relies entirely on standard, publicly available image classification datasets and does not involve any crowdsourcing or human subjects.

14. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The research does not involve human subjects, so IRB approval is not applicable.

15. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [N/A]

Justification: LLMs are not used as a component of the core methodology, algorithm design, or experimental evaluation in this research.